Prediction of local concentration fields in porous media with chemical reaction using a multi scale convolutional neural network

Chemical Engineering Journal xxx (xxxx) xxx

Contents lists available at ScienceDirect

Chemical Engineering Journal

journal homepage: www.elsevier.com/locate/cej

Prediction of local concentration fields in porous media with chemical reaction using a multi scale convolutional neural network

Agnese Marcato [a,*], Javier E. Santos [b], Gianluca Boccardo [a], Hari Viswanathan [b], Daniele Marchisio [a], Maša Prodanović [c]

[a] Department of Applied Science and Technology, Politecnico di Torino, Torino, Italy
[b] Earth and Environmental Science Division, Los Alamos National Laboratory, NM, USA
[c] Hildebrand Department of Petroleum and Geosystems Engineering, The University of Texas at Austin, TX, USA

ARTICLE INFO

Dataset link: https://github.com/mulmopro/reactive_ms_net

Keywords:
Porous media
Convolutional neural networks
Machine learning
CFD
Reaction
Filtration

ABSTRACT

The study of solute transport in porous media is of interest in many chemical engineering systems. Some example applications include packed bed catalytic reactors, filtration devices, and batteries. The pore scale modeling of these systems is time consuming and may require large computing resources, for this reason computational fluid dynamics (CFD) simulations are not practical if a large number of simulations is required, like in multiscale modeling, where a model at a large scale calls for pore scale simulations. It has been shown that neural networks can be trained with a dataset of flow simulations and then predict fields orders of magnitude faster, and with less computational resources, in new domains. However, it is crucial to provide the neural network with an effective description of the domain and the undergoing operating conditions to be able to train models that generalize accurately in unseen samples. Therefore, research is needed to employ neural networks in new complex systems. The appropriate training of a network for predicting coupled flow and solute transport processes is an outstanding problem due to the complex interplay between geometry and operating conditions. In this work, we train a multi scale convolutional neural network (MSNet) with a diverse dataset of simulations of transport and chemical reaction in porous media to predict the local concentration fields in images of porous media. Our dataset contains a wide diversity of sphere pack arrangements under different operating conditions (Péclet and Reynolds numbers). We train a robust model by employing different input descriptors that represent the medium and the different operating conditions of each system. Our trained model is able to provide nearly instantaneous predictions, compared to around twenty hours of the CFD workflow, with less than 3.5% error on new geometries and transport conditions. Thus the model could be easily integrated in a multiscale workflow where fast response is needed.

1. Introduction

Computational fluid dynamics (CFD) is a well-established modeling approach for flow and transport in porous media. Some classical applications in chemical and environmental engineering involving the modeling of porous media systems include packed bed chemical reactors [1–4], aquifer remediation [5,6], subsurface flows [7], filtration [8–11], and separation devices [12,13]. In the last decade the contribution of chemical engineering research towards new energy solutions has been crucial in the field of carbon capture and storage [14–17], and batteries [18,19], where, at different scales, rocks and electrodes are modeled as porous media. All these relevant applications require the modeling of complex phenomena through CFD simulations, that provide pore scale accurate solutions of the transport quantities of interest.

Depending on the complexity of the physical problem, and on heterogeneity and size of the domains, both meshing and the subsequent simulation can be highly time consuming and computationally expensive to run. For this reason, the use of high-performance computing (HPC) systems is often required to solve the simulations in parallel.

When numerous simulations are needed (like in optimization algorithms and in multiscale modeling [20–22]) or when real-time predictions are necessary (like in-line plant control [23]), it would be extremely useful to have fast and accurate models to predict the system behavior (or performance) based on locally changing microscale conditions. Given the clear multi-scale nature of porous media transport phenomena, historically a lot of effort went into the development of such tools, differing widely in approach, from theoretical

* Corresponding author.
E-mail address: agnese.marcato@polito.it (A. Marcato).

https://doi.org/10.1016/j.cej.2022.140367
Received 11 July 2022; Received in revised form 16 October 2022; Accepted 11 November 2022
Available online 15 November 2022

upscaling approaches to the development of phenomenological constitutive equations.

An example of the first class of solutions, aside from well known averaging procedures [24] is the analytical development of models by means of asymptotic homogenization, which has enjoyed great success in obtaining closed forms of macroscale transport equations [25,26] but which suffer (due to the complicated analysis involved) in limits to its applicability both in treatable geometrical structures [27] and transport regimes [28]. Other approaches are based on building constitutive equations from both empirical or computational results and while they have been vastly employed in many different fields [29–31], these relations are still prone to fail when the geometries become random [32] and are hardly parametrizable [33]. Many solid research works have thus focused on simpler models [34,35] and even slightly more complicated pore/collector geometries have been found to noticeably complicate things in terms of obtaining upscaled laws [36].

Then, one alternative to the mentioned approaches, which is gaining momentum in the last few years, is to train specific neural network models in order to obtain these fast response surrogate models. However, as of now, problem-specific design choices have to be made to identify the most suitable neural network architecture for each problem to be solved.

In the porous media research, different kinds of neural networks have been trained on datasets of physics-based simulations. Fully connected neural networks have been employed for the prediction of integral quantities, such as the permeability, from effective, hand-picked features (such as the porosity or the surface area). This approach can be found in literature [37,38], and in our earlier works for the study of flow and transport in porous media [39,40]. These techniques are effective and their training is easy to carry on, but the resulting model is very sensitive to the choice of input parameters, since the domain is described via upscaled parameters (say, porosity) that have no way of relaying spatial heterogeneity. This choice relies both on user expertise, and at least as importantly on their accurate calculation from the system under investigation. Therefore, these models tend to fail in heterogeneous domains (since the effective features are an oversimplification) and cannot easily be transferred to a new dataset.

The increasing availability of GPUs (graphics processing units) and open-source deep learning libraries have made the training of more complex neural networks computationally feasible. The use of deep learning techniques such as convolutional neural networks (CNN) ease the choice of the right integral descriptors of the porous media since the entire system geometry is fed to the network as an image, and the network autonomously detects the most effective features for the prediction of the objective output. It has to be remarked that this does not result just in an automatic choice of relevant integral features, but in a trained network that operates by "seeing" the system geometry and is then able to make predictions based on the experience thus acquired. Most of the works in the porous media field employed these techniques to predict the medium permeability [41,42], whereas in our latest work physics informed neural networks were employed to predict also other relevant quantities in process engineering, such as the filtration rate (or reaction rate) in porous media over a range of certain operating conditions [43].

Given this context, what seems a natural evolution of this research is to build CNN that can be employed to surrogate the microscale local solution of the transport equations too. In fact, CNN with encoding and decoding architectures have been used to train surrogate models able to predict the flow field in different microscale porous media systems [44–46] and for uncertainty quantification in macroscale subsurface applications [47–51]. Our latest architecture, Multi Scale Neural Network (MSNet) [52] came out to be a preferred alternative to the previous ones, from both the computational point of view and, notably, its generalization capability. In fact, it is possible to train the network with larger geometric samples than what more classical approaches allow, which is fundamental when dealing with representative elementary volumes of heterogeneous geometries and/or complicated transport phenomena. This was possible thanks to the employment of a series of different fully convolutional neural networks that focus on different resolutions of the input features, each contributing to the final prediction. This feature makes it able of understanding, and correlating, transport phenomena at different length scales. In turn, this capability of grasping the correlations between different scales is what makes the network able to predict the flow fields in new geometries, different from the ones learnt during the training.

Nonetheless, as the main effort in the last years was addressed to the development of architectures for the microscale prediction of flow fields, little was done to expand these methodologies to more complex physical systems, which are of common experience in the chemical engineering field.

The complexity does not arise only from the different prediction objectives, but also from the kind of input necessary for the network. When the main objective is the prediction of the permeability, as in the above-mentioned studies, it can be evaluated from the prediction of the flow fields. In laminar flow regime the permeability is just related to the geometry of the porous media [53,54], so a dataset at constant pressure drop is sufficient. Instead, in coupled flow and solute transport problems, different pressure drops (or other operating conditions) impact the ultimate solute concentration field even if the flow regime does not change. In this work we provide proof of the capability of these CNN models to surrogate CFD simulations of transport and reaction under a wide range of different transport conditions that are common in chemical engineering problems.

Thus, the objective we set for this work was to obtain a network able to predict concentration fields in porous media in the case of a heterogeneous surface reaction (or equivalently filtration) when dealing with a wide range of Reynolds and Péclet numbers. To this end, we modified the original MSNet and trained it with a "ground truth" dataset of CFD simulations of flow and reactive transport over many geometrical realizations and different operating conditions. The concentration fields to be predicted depend on the geometry, on the pressure drop between input and output of the porous media, and on the diffusion coefficient of the chemical species transported. Beyond only having a working neural network for the solution of this problem, we also propose a wide-ranging study of the different possible input features employable, resulting in the best set of features that allows the network to satisfactorily generalize its predictions to new geometries and operating conditions.

For the interested reader (and to give a glimpse of the possible application of the results we present in this work) we refer to the first Section of the Supporting Information document. There, we detail a specific multiscale problem (prediction of colloid filtration in different porous media structures and under different conditions) and how specifically this approach may help when the available constitutive equations fail.

2. Methods

In this section the governing equations and the computational setup of the CFD simulations are detailed. Then the approach for the creation of the dataset is described. Finally, a theoretical background about CNN and the details about MSNet are reported.

2.1. Computational fluid dynamics simulations

The flow in porous media at the microscale, or pore scale, is governed by the continuity equation, Eq. (1), and the Navier–Stokes equation, Eq. (2). Under the hypotheses of incompressible and Newtonian fluid those read as follows:

$$\frac{\partial U_i}{\partial x_i} = 0, \tag{1}$$

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j^2}, \tag{2}$$

where U_i is the i^{th} component of the velocity, p is the pressure, ρ is the fluid density, and ν is the kinematic viscosity.

The scalar transport of a species can be modeled by the advection–diffusion equation, Eq. (3), if the species is considered to move with the fluid at the same velocity. In this case the diffusion can be described by the Fick's law, and the equation reads:

$$\frac{\partial C}{\partial t} + U_i \frac{\partial C}{\partial x_i} = \frac{\partial}{\partial x_i}\left(D \frac{\partial C}{\partial x_i}\right), \tag{3}$$

where C is the species concentration and D is its diffusion coefficient. The modeling of transport by the advection–diffusion equation holds for chemical species and colloids under the hypotheses of dilute system and negligible particle Stokes number. In fact, colloids filtration modeling by using the previous equations was explored together with a detailed presentation of the validity of the hypotheses by Boccardo et al. [32].

These equations were numerically solved by the finite volume method implemented in the CFD open-source code OpenFOAM v7. First the continuity and the Navier–Stokes equations are solved, then the resulting velocity field is employed for the solution of the advection–diffusion equation, so two subsequent CFD simulations are run to obtain the steady-state concentration field — which is the prediction objective of this work.

The geometries created for this work are bi-dimensional: given a chosen porosity and their diameter, a number of non-overlapping circles are placed in a square while respecting a periodic constraint at the top/bottom boundaries of the image, Fig. 1.

The number of circles was chosen in order to obtain a representative elementary volume for this geometry, following the results from our previous work [40]. The mesh was computed by means of the OpenFOAM tools `blockMesh` and `snappyHexMesh`.

The fluid considered is water at room temperature with density equal to 997 kg m^{-3} and kinematic viscosity equal to 0.89 $\times 10^{-6}$ m^2 s^{-1}. The SIMPLE algorithm implemented in the solver `simpleFoam` was employed for the coupled solution of the continuity and the Navier–Stokes equations in steady-state and laminar conditions. In order to obtain a system with an identifiable main flow direction aligned with the Cartesian axes (i.e. left–right direction, see Fig. 1), boundary conditions resulting in a pressure drop between input and output of the porous medium are set, together with zero gradient conditions for the velocity. Then, the impermeability of the circular objects was expressed by no-slip velocity conditions and null gradient for pressure on the circles surface. Periodic flow boundary conditions were set on the remaining boundaries.

The advection–diffusion equation was solved in steady-state conditions by means of the solver `scalarTransportFoam`. The boundary conditions set were a normalized unitary inlet concentration, a null concentration on the grains surface, a null gradient conditions at the outlet, and periodic conditions on the remaining boundaries. The null concentration on the circles surface was set in order to represent an instantaneous reaction of a chemical species on the surface; alternatively, this model is also appropriate to represent the problem of filtration of colloidal particles in the case of favorable colloid-filter interaction. The reader interested to the latter is referred again to our previous work [32], as a bridge to the relevant literature on colloid filtration theory. In S2 of the *Supplementary Information* the numerical setup of the CFD simulations is detailed.

2.2. Creation of the dataset

The training of the neural network requires a dataset that encompasses a wide variety of geometries and operating conditions, in this work we carried out 800 CFD simulations of flow and transport in porous media, all with different operating conditions and domain geometries. We have chosen to perform this quite extensive number of simulations precisely for the purpose of exploring the effect of the size of the training dataset on the neural network performance.

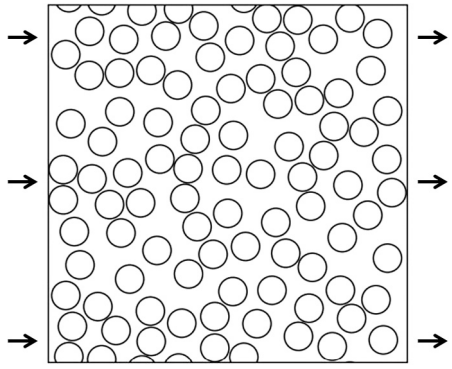

Fig. 1. Bi-dimensional porous medium geometry created by setting a periodic constraint at the top/bottom boundaries.

Table 1
Range of variation of the features chosen for the creation of the geometries and the solution of the CFD simulations.

Parameter	Range of variation	
p	0.30–0.50	Pa
d_g	100–200	μm
ϵ	0.5–0.65	(–)
$\mathcal{D}$	3.13×10^{-11}–5.71×10^{-10}	$m^2 s^{-1}$

The samples of the dataset are chosen so to explore a wide range of diameter of the circles, porosity of the 2D packing, pressure drop across the porous medium, i.e. the Reynolds number, and diffusion coefficient, i.e. the Péclet number, Eq. (4):

$$\mathrm{Pe} = \frac{q d_g}{D}. \tag{4}$$

where q is the superficial velocity, and d_g is the diameter of the grains (circles in this 2D case). These features range of variation is listed in Table 1.

The mesh created for the CFD simulations is stair-stepped (*castellated* in OpenFOAM parlance) in order to ease the use of the resulting concentration fields for the training of the neural networks without needing an interpolation from a body-fitted mesh (with non-Cartesian structure) to the matrix-like data structure needed by the neural network training calculations.

Then, a grid independence study was performed in order to choose the size of these computational grid elements, by monitoring the medium permeability and the average concentration in the domain with varying mesh cell size. Since in the dataset of CFD simulations a wide range of operating conditions was explored, the grid independence study was performed on the sample subjected to the highest Péclet number, which is the most critical condition from the computational point of view, due to the smaller boundary layer on the surface of the grains caused by the null concentration boundary condition. In Fig. 2 the study is reported, with the green line highlighting the chosen mesh strategy, corresponding to a linear discretization of the square domain of 1536 cells. The relative error between the average concentration and the permeability calculated from the simulations performed with the chosen meshing strategy and the most refined grid tested is, respectively, 0.5% and 0.1%. The results of the chosen strategy can thus be considered fully grid independent.

2.3. Convolutional neural networks

CNN are a class of neural networks suited to deal with grid-like objects as input features [55], like images. The CNN layers implement

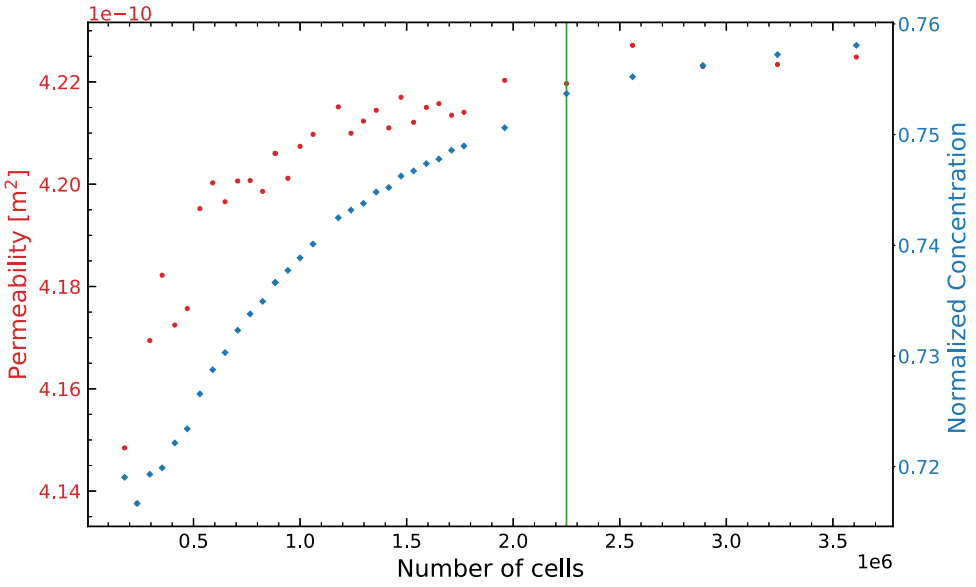

Fig. 2. Grid independence study performed for the CFD simulations. The average concentration in the domain (blue diamond markers) and the permeability (red bullet markers) were monitored. The green line indicates the grid independent result whose strategy was chosen.

the convolution operation:

$$y = f\left(\sum_{i=1}^{F} x * k_i + b_i\right), \tag{5}$$

where $*$ denotes the convolution operation, x is the input, y is the output of the operation, f is the activation function, k_i is the kernel, F is the number of kernels, and b_i is the bias term. The kernel is a trainable array of floating-point numbers of a chosen size that is applied on the image. In this work the input features are images, so the filter is two-dimensional of size 3-by-3, which is the most computationally efficient size for GPU computations [56].

CNNs have exhibited excellent performance in deep learning tasks compared to classical fully connected neural networks both in terms of generalization capability and computational cost of the training [57]. This is partly due to the fact that convolutional kernels share their parameters, because the same filter slides on the image and is applied in different regions, thus every output is connected just on a small portion of the input, which is referred to as sparse connectivity. Consequently, CNN are characterized by equivariance to translation, so the layers learn the influence of the images features independently from their location.

The training of neural networks aims to optimize the trainable parameters of the network to minimize the loss function, which is done with the back-propagation algorithm. In this work we used the `Pytorch` [58] implementation of Adam [59], which is one of the most employed and stable gradient descent optimizers [60].

As with other deep learning algorithms, CNN may suffer of gradient vanishing issues [61], thus the choice of an appropriate activation function is crucial to assure the contribution of each neuron to the final prediction. The selection of the activation function is one of the many choices to be made for the design of the neural network model.

The most used activation functions nowadays are the Exponential Linear Units (ELUs), nevertheless activation functions like the Gaussian Error Linear Unit (GELU) seems to be a promising alternative for convolutional networks [62]. For this work we tested both the use of a ELU, the continuously differentiable exponential linear unit (CELU), and GELU (the reader is referred to the *Supporting information* section for a deeper discussion about this comparison). Since there was no apparent impact of the activation function on the accuracy of the predictions for our trainings, we decided to use CELU, the original and widely tested activation function of MSNet. The continuously differentiable exponential linear unit (CELU) reads as follows [63]:

$$\mathrm{CELU}(x) = \max(0, x) + \min\left(0, \alpha \cdot \left(e^{\frac{x}{\alpha}} - 1\right)\right) \tag{6}$$

where α is a parameter controlling saturation for negative inputs [64]. CELU avoids saturation of the output unlike other activation functions traditionally employed in neural networks, such as the sigmoid function, and avoids the *dead ReLU* problems being an ELU activation function [65].

2.4. MSNet: Multi Scale Neural Network

In this section the main characteristics of MSNet are summarized. This architecture was originally presented by Santos et al. [52], for a detailed description the reader may refer to this work.

MSNet is a convolutional neural network structured in scales, where each scale aims to predict the property field structure at a different level: a sketch of the mechanism is presented in Fig. 3. The architecture has two purposes: (1) letting each scale focus on different length scales of the field (i.e. the scale with the largest field of vision can capture the global trend of the field) and (2) allowing to train computationally large images in a single-GPU (which is not feasible with a model such as U-Net [66] or Res-Net [67]). The scale that deals with the full domain size is denoted as *scale 0*, subsequent scales (1 through N) receive the same input as the previous scale but coarsened by a factor of two. The last scale N receives the coarsest representation of the input feature, consequently, its output is the coarsest representation of the predicted field. This output is refined and fed to the previous scale together with the refined input feature. The output of the $N-1$ scale is summed with the refined output of scale N. This procedure is replicated for all the intermediate scales, and can be summarized as follows:

$$\hat{y}_{N-1} = \mathrm{CNN}_N(X_N, \mathbb{R}(\hat{y}_N)) + \mathbb{R}(\hat{y}_N), \tag{7}$$

where $\hat{y}_N$ is the predicted field at scale N, CNN_N is the fully convolutional neural network of scale N, X_N is the set of input features coarsened by a factor 2^N, $\mathbb{R}()$ is the refinement operation.

The coarsening of the input features and the output fields is performed by a nearest neighbor averaging, meaning that the value of every 2^2 group of pixels is averaged into a single pixel. Instead, the refinement procedure consists in a masked nearest-neighbors re-scaling that maintains the shape of the geometry of the solid portion. For a

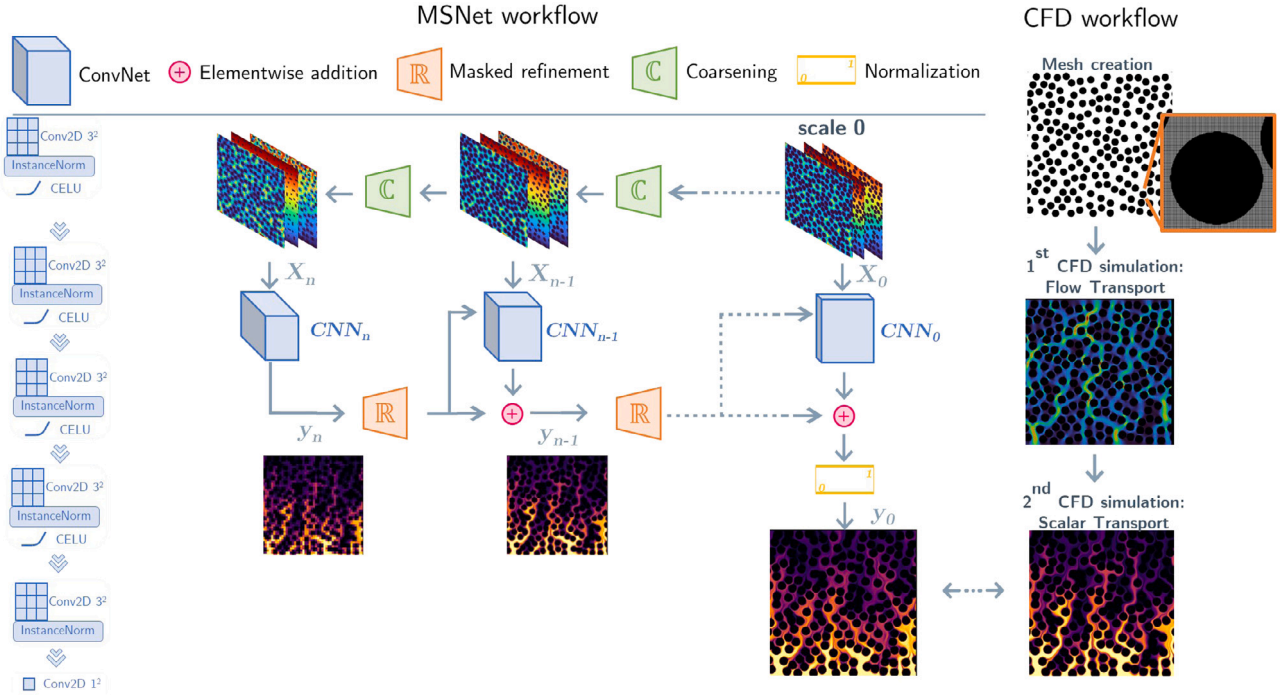

Fig. 3. MSNet workflow (left) and CFD workflow (right). A set of input features is chosen for the prediction of the steady-state concentration field (presented in rainbow color scale in the figure). The input features tested are the Euclidean distance transform, the time of flight, the local thickness, the operating pressure drop, and the diffusion coefficient. Each scale is made by a convolutional network block that is detailed on the left. The coarsening and masked refinement operations are employed to transfer information between scales. The CFD workflow is summarized on the right. The output of the simulations is the ground truth for the training of neural network.

more detailed explanation of the operation the reader may refer to Section 2 of the original MSNet work [52]. The coarsening and refinement operations conserve the average over space, and the coarsening of a refined image gives back the original image. It is important to notice that the coarsening operation is performed just once before the training in order to calculate the coarse fields of the input features and of the true concentration fields, that will be employed during the training as ground truth at each scale. The time required to perform the refinements is not rate determinant during the training. Just 0.8% of the time of the training is spent in the refinement calculations.

The loss function employed in the training of MSNet considers the prediction error of each scale. The contribution of each scale is given by the mean squared error between the predicted field $\hat{y}$ and the true field y (coarsened at the corresponding level) divided by the variance of the true field, $\sigma^2_{y_i}$, so the global loss function L is:

$$L = \sum_{s=0}^{N} \sum_{i=0}^{N_S} \frac{\left\langle \left(y_{i,s} - \hat{y}_{i,s}\right)^2 \right\rangle}{\sigma^2_{y_i}}, \tag{8}$$

where the index s refers to the scale, i refers to the sample, N_S is the number of samples in the training set, the operator $\langle\cdot\rangle$ refers to the spatial average on the domain.

The architecture of the neural network is the same for each scale and is summarized in Fig. 3. The networks are fully convolutional, and the size of the images is maintained along the layers of the scale. The first four blocks are made by a convolutional layer whose kernel size is 3^2, followed by a normalization layer, after that the CELU activation function is applied. A fifth block without activation function is then added in order to not constrain the output. The number of kernels of each convolutional layer depends on the scale: 2^{2s+1}. Finally, the network ends with a convolutional layer with a single kernel size of 1^2 in order to reduce the dimensionality of the output to a single image, which is the predicted field.

Conceptually MSNet can grasp both short-range and long-range correlations in the field thanks to the use of the same features at different resolutions, resulting in good generalization capability for the flow field prediction, and the related permeability prediction. From the computational point of view, using the same input features at different resolutions allows for a higher number of kernels at the coarsest scales and a lower number of them at the finest scales: as a consequence, the training is faster, and the memory requirements are decreased.

2.4.1. MSNet for the prediction of concentration fields

The MSNet architecture was originally conceived for the prediction of flow fields. Since the main objective was the prediction of the permeability, a dataset of simulations in laminar conditions at constant pressure drop was employed [52]. As mentioned, while permeability is strictly related to fluid flow, it is determined only by the geometric features of the porous medium [53,54]: therefore, this geometrical description was the only input feature needed by the network for the prediction of the field. In this work the architecture was employed for the prediction of concentration fields, and the main effort was headed in the choice of the most appropriate input features. In this section the input features tested are described and information about their extraction is provided, while their effectiveness on the prediction of the concentration field is discussed in the following section. In the proposed dataset both the porous media geometries and the operating conditions of transport affect the concentration fields, so the neural network must be provided with both features.

Concerning the geometrical description of the porous media the features tested are: the Euclidean distance transform, the linear variation distance from a boundary, the time of flight, and the local thickness, Fig. 4.

The Euclidean distance transform is applied to the binary images of the porous media, where 0 labels the solid phase and 1 the fluid phase. As a result, for each pixel of the fluid phase the Euclidean distance is calculated from the closest solid pixel, i.e. the closest solid grain, Fig. 4A.

As stated in the previous section, CNN are invariant to translation, but the concentration values in the pore space are linked to their

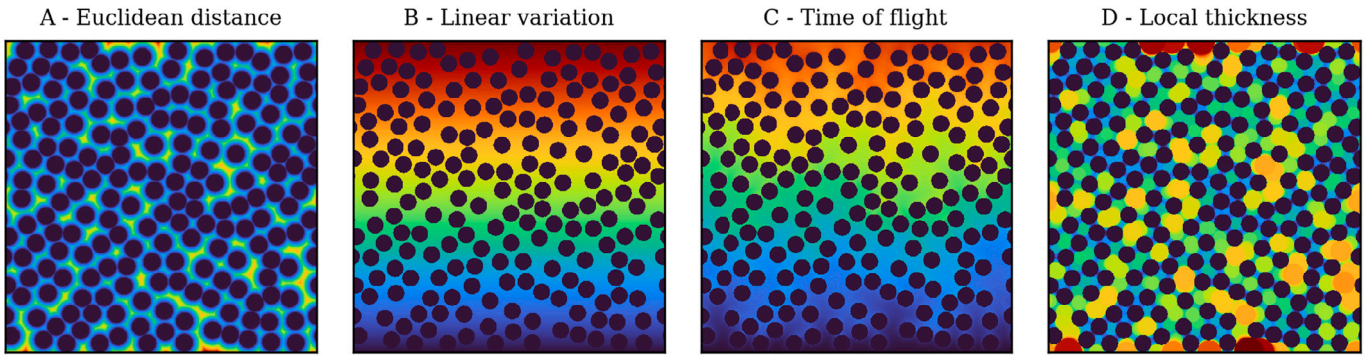

Fig. 4. Geometrical input features tested as input to MSNet for the prediction of the concentration fields. A — Euclidean distance transform of the binary image; B — Linear variation of the coordinate in the main flow direction; C — Time of flight in the main flow direction; D — Local thickness.

position with respect to the inlet boundary. Even though MSNet can grasp the spatial correlations at different scales, being the transport of species concentration a strictly oriented phenomenon, it is necessary to provide the neural network with this information. It can be conveyed by a simple linear variation of the coordinate in the flow direction, Fig. 4B, or with more informative features that can account for the tortuosity of the porous medium.

The time of flight describes the tortuosity in the porous medium as the shortest distance of a point from a chosen boundary, Fig. 4C. In order to calculate it a boundary value problem of the Eikonal equation is solved:

$$F(x)|\nabla(t(x))| = 1, \tag{9}$$

where $F(x)$ is the speed at which the boundary x evolves in time t. The fast marching method implemented in scikit-fmm [68] was employed in this work to compute the time of flight.

The boundary selected is the inlet boundary of the transport simulations, and the speed field is the Euclidean distance field, Fig. 4A. Commonly the speed field is set with a constant value in the pore space and zero in the solid phase: however, the resulting time of flight field does not highlight the preferential paths and a mostly linear variation of the coordinate is returned, since the circular obstacles have a negligible influence on the boundary path. If a non-constant speed field as the Euclidean distance is employed, it is possible to extract a more informative feature describing the preferential paths in the porous media that are of the utmost importance in the shape of the concentration fields.

Other geometric features have been taken into consideration to aid the neural network for the non-trivial prediction of the concentration field, such as the local thickness, i.e. maximum inscribed sphere, Fig. 4D. To obtain this feature, we used the approach implemented in PoreSpy [69] which consists in finding which group of pixels can accommodate a sphere of a given radius. Given the number of sizes and the bins in the size distribution, the algorithm detects the largest pore applying the Euclidean distance transform to the image. Then the algorithm searches for all the smaller spheres between zero and the largest one by using the Fast Fourier Transform convolution.

All the above-mentioned features are not computationally expensive, the time required for their calculation is in the order of a few seconds per sample. Consequently, the choice of these features can easily scale to a three-dimensional dataset.

The operating conditions characterizing the dataset are the pressure drop across the porous media, and the diffusion coefficient of the chemical species transported. The latter is provided to MSNet as an image with the value of the diffusion in each pixel of the pore space. The pressure drop feature can also be supplied to the network in this way, like the diffusion coefficient, or it can be merged with the information about the distance from the inlet boundary (Fig. 4B) since the pressure decreases along the flow direction. Scaling the features of this kind (linear variation and time of flight) by the value of inlet pressure allows for a more compact set of features to pass to the CNN. Lastly, a unique feature for the operating conditions such as the ratio between the pressure drop and the diffusion coefficient was tested in this work too.

Concerning the scaling of the input features, those are scaled by their mean value over the whole dataset. In the case of the output concentration field instead, no further scaling is needed, as it is already normalized between 0 and 1 from the CFD simulations.

Since the output concentration is always between 1, at the inlet boundary, and 0, at the grain surface, a normalizing layer was added at the output layer of MSNet to prevent the prediction of nonphysical results.

The number of scales chosen for the MSNets trained in this work is 6, which represents a good tradeoff between limiting the number of trainable parameters and obtaining a well coarsened representation at the last scale. In fact, the field of vision (FoV), i.e. the number of pixels of the input affecting each pixel of the output, is defined as follows for MSNet:

$$\mathrm{FoV}_{\mathrm{MSNet}} = (L(k_{size} - 1) + 1)2^n, \tag{10}$$

where L is the number of convolutional layers of the network (5 in this MSNet), k_{size} is the size of the kernels (3 in this MSNet), and n is the number of scales. Given the size of the CFD simulations fields (1536×1536), the FoV of the finest scale ($n = 0$) is equal to 11 voxels and the one of the coarser scale ($n = 5$) is equal to 352. We performed the trainings on NVIDIA Volta V100 GPUs, Nvlink 2.0, 16 GB.

3. Results and discussion

3.1. CFD simulations for the creation of the dataset

The neural network for the prediction of concentration fields is trained on a dataset made by CFD simulations. Two CFD simulations are solved to obtain the concentration fields, as summarized in the workflow of Fig. 3. At first the velocity field is obtained from the coupled solution of the continuity and Navier–Stokes equations, after that the scalar transport is simulated by the solution of the advection–diffusion equation.

Each complete simulation constitutes a sample point of the dataset. The total number of simulations solved is larger than the minimum set size required for the training of an MSNet for concentration prediction: this was done to perform a sensitivity analysis on the number of actual samples required to obtain a satisfactory accurate prediction.

An HPC cluster was employed to solve many (single core) simulations at the same time. In this way, 800 simulations have been solved. The time required to create the castellated mesh, and to solve the two CFD simulations is about 20 h on an HPC cluster equipped with 29

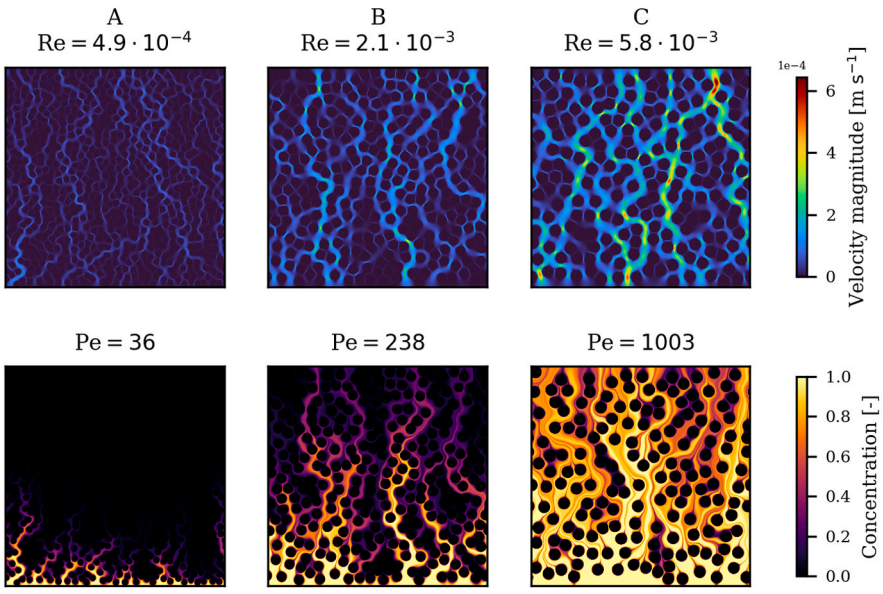

Fig. 5. Contour plots of the velocity and concentration fields for three samples of the dataset. A lowest Péclet number — the diffusive term prevails. B intermediate Péclet number. C highest Péclet number — advection term prevails, the chemical species flows easily through the porous medium.

nodes with CPU 2x Intel Xeon E5-2680 v3 2.50 GHz 12 cores RAM of 384 GB.

A random combination of the input features in the range displayed in Table 1 results in a range of Reynolds numbers of 3.96×10^{-4}–1.58×10^{-2}, and in a range of variation of the Péclet number of 23–1500. Thus, a wide range of transport conditions in laminar flow was explored in the dataset resulting in a challenging dataset for MSNet.

In Fig. 5 the flow field and the concentration fields for three samples of the dataset are shown. The lower is the Péclet number, the higher is the contribution of diffusion with respect to the convective term (which regulates the residence time of the chemical species in this reactive system). As a result, the average concentration in A is lower than the average concentration in B, which is lower than the average concentration in C.

3.2. Neural networks

The choice of the input features for the prediction of the concentration field is of the utmost importance for the generalization capability of the neural network.

The prediction accuracy of the networks is visually depicted with the fields of local errors between the predicted concentration field and the CFD result. In addition, three metrics are employed in order to easily compare the prediction accuracy on the test set:

- The percentage error on the average concentration of the field;
- The root mean squared error (RMSE) of the concentration profiles in the flow direction;
- The RMSE of the concentration profiles in the direction perpendicular to the flow.

The result of these metrics for each sample of the test set is averaged in order to obtain a metric for the entire test set.

In Table 2 the different combinations of input features tested are summarized, and the metrics previously described are reported for each set of features. In case A, just the basic features are provided, so the operating conditions (pressure drop and diffusion coefficient) and the geometrical conditions, by means of the Euclidean distance, Fig. 4 A. The three metrics show the highest error, in fact, the predicted field is not physical since there are zones in the fluid where the concentration increases moving towards the outlet boundary; in a filtration, or depleting reaction problem, this obviously should not happen (Fig. S2 of the *Supporting information)*. This happens because of the lack of information about the distance from the inlet boundaries: in fact, the convolutional layers are invariant to translation, so it is necessary to provide the network with the position with respect to the inlet boundary.

If the distance from the inlet boundary is given as a coordinate linear variation, Fig. 4B, normalized by the pressure drop, case B, the accuracy remarkably improves. The error on the average concentration in the fields decreases from 13.5% to 5.7%, the RMSE for the concentration profile in the flow direction decreases from 0.104 to 0.034, and the RMSE for the concentration profile in the perpendicular direction to flow decreases from 0.074 to 0.054.

The time of flight, Fig. 4 is an alternative option to provide the information about the distance from the inlet boundary. Compared to the linear variation of the coordinate, the time of flight embeds the description of the tortuosity in the geometry and underlines the presence of preferential paths. If both features (time of flight and linear variation of the coordinate) are fed to the network, case C, the improvement of the generalization capability does not increase in a relevant manner.

On the other hand, if just the time of flight is employed, together with the Euclidean distance, the diffusion coefficient and the pressure drop, case D, the accuracy remarkably increases. This results in an error on the average concentration of 3.3%. The redundant information about the distance is not useful for the network, so it is preferable to just provide the network with the time of flight.

Then, since it is desirable to reduce the number of input features, in order to decrease the memory load on the GPU and the number of trainable parameters, and as a consequence the computational cost of the trainings, the compression of the two operating conditions features into a single one was tested. Employing the ratio of pressure drop and diffusion coefficient, case E, the prediction accuracy remains unchanged, so this solution is preferable to the previous one.

Following the same approach, a training without the Euclidean distance feature was performed in order to reduce the number of input features to the minimum, case G. In this case the generalization capability is not preserved, in fact, the error on the average concentration increases again from 3.3% to 5.5%. Thus, the Euclidean distance cannot be missed in the set of features for the prediction of the concentration fields.

The last set of features tested is the same of case D but with the additional feature of the local thickness, case F. The new geometrical

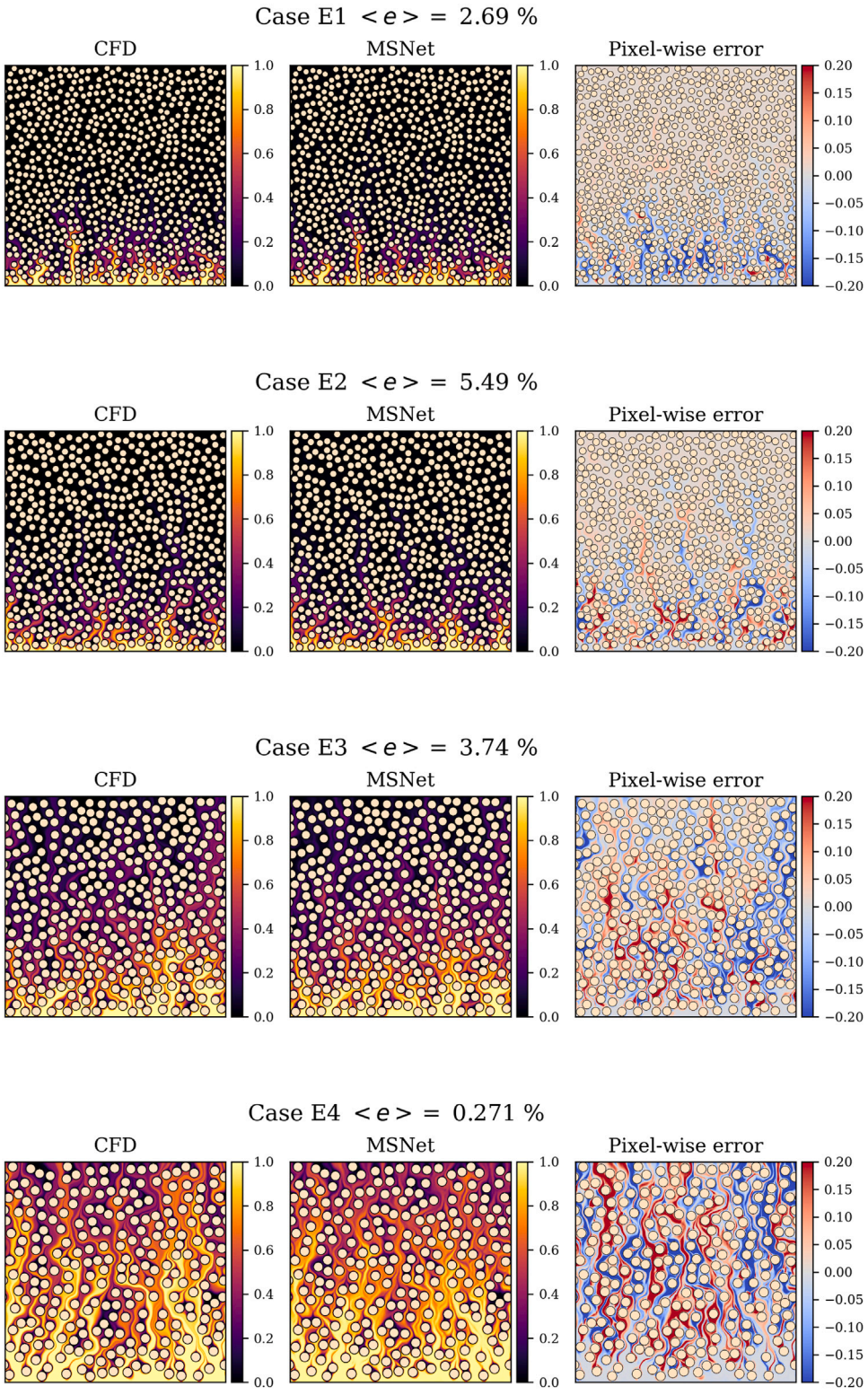

Fig. 6. Prediction of MSNet for four samples of the test set. From left to right: concentration field resulting from the CFD simulations, concentration field predicted by MSNet with the input features of case E Table 2, field of local error computed as the pixel-wise difference between the CFD result and the MSNet prediction.

feature does not improve the accuracy, so it is not worth it to add local thickness to the set of input features.

Given the presented results, the best set of features was found to be the one from case E, where the operating conditions are provided in a single feature in the form of the ratio of the pressure drop and the diffusion coefficient, and the geometric features into two features, the Euclidean distance and the time of flight. This last feature is particularly effective since the preferential paths information is clearly

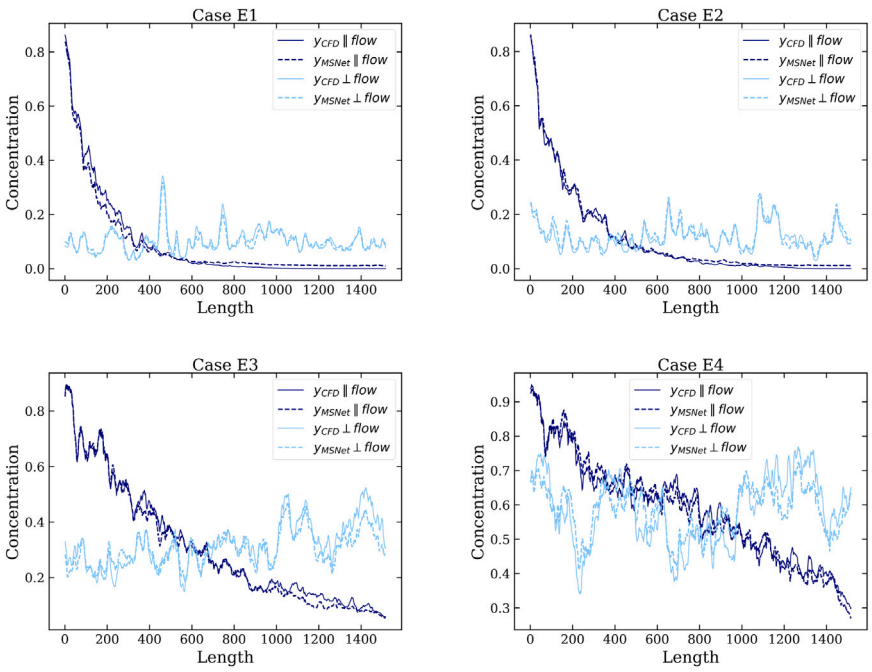

Fig. 7. Profiles of average concentration in the direction of flow and in the perpendicular direction to flow (CFD: solid, MSNet: dashed) for the four samples of Fig. 6.

correlated to the concentration field patterns. The results of the training of MSNet with the best set of features are reported in Fig. 6, where the predictions for four samples of the test set are shown. The percentage error on the average concentration between the result of the CFD and the prediction of MSNet is in the label of each sample. One point of note is that, beyond quantitative error measure, the network is can predict qualitative transport features with satisfactory precision, for example the low-concentration tails in the wake of the grains.

Since the samples in the test set differ from those in the training set for all the input features, including circles diameter and most importantly their placement, it is significant to note that these results describe the performance of this model on a generalized form of this transport and reaction problem. If this model were employed on a similar system (one never seen in the training of the network), similar performances to those reported would be expected.

In Fig. 7 the profiles corresponding to the fields of Fig. 6 represent an alternative way to evaluate the prediction error of the network. Also in this case, it can be appreciated that the average profiles predicted by MSNet quite closely follow the shape of the profiles calculated from the results of the CFD simulations.

It is important to notice that the accuracy of the results obtained is deeply connected to the multiscale approach of the network. The architecture of the network permit to use a larger number of trainable parameters compared to other CNN of common use. We compared the prediction accuracy of MSNet with a fully convolutional neural network and it results that, using the same GPU, the amount of trainable parameters that we can fit into the GPU is lower than MSNet (just 1.3 million instead of 161 million), and the accuracy of the predictions is highly decreased. These results are compared in section S5 of the *Supporting information*.

A sensitivity analysis on the dimension of the dataset was performed in order to detect the minimum number of samples necessary to achieve the accuracy presented previously. An increasing number of samples are employed during training, from 200 to 800 (the entire dataset). In Fig. 8 the results are reported, for each dataset size the training was reproduced three times. It is possible to conclude that at least 400 samples are required to achieve an accuracy on the average concentration of around 3.5%. The test set used to evaluate the generalization capability of the networks was the same. The use of a specialized neural network architecture and of appropriate input features, both conceived for the prediction objective, is essential to decrease the amount of samples necessary to obtain a satisfactory accuracy.

Table 2
Input features tested for the training of MSNet. The different combinations are compared on the error on the prediction of the average concentration, on the RMSE of the concentration profiles in the flow direction and in the perpendicular direction to flow.

	A	B	C	D	E	F	G
Linear variation normalized by the pressure drop		✓	✓				
Pressure drop	✓			✓		✓	
Pressure drop/Diffusion coefficient					✓		✓
Diffusion coefficient	✓	✓	✓	✓		✓	
Euclidean distance	✓	✓	✓	✓	✓	✓	
Time of flight			✓	✓	✓	✓	✓
Local thickness						✓	
$\langle e \rangle_{test}$ on average concentration	13.5%	5.7%	5.3%	3.3%	**3.3%**	5.3%	5.5%
$RMSE_{test}$ on concentration profile parallel to flow	0.104	0.034	0.027	0.023	**0.024**	0.032	0.027
$RMSE_{test}$ on concentration profile perp to flow	0.074	0.054	0.042	0.042	**0.042**	0.044	0.044

4. Conclusions

Neural networks, both fully connected and convolutional, have been widely employed to train data-driven models that quickly reproduce the results of more computationally expensive physics-based simulations.

In this work a multi-scale convolutional neural network was trained to reproduce the full concentration profile of different samples, which is commonly obtained via CFD simulations. This approach differs from other recently proposed machine learning workflows by being able to predict the entire concentration field of a large image, instead of just a

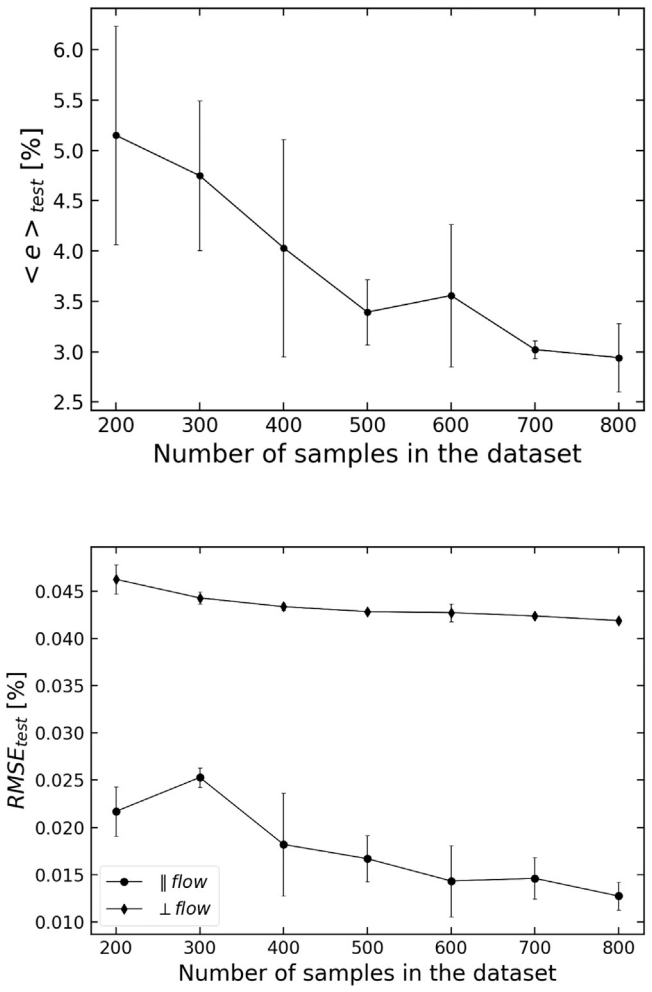

Fig. 8. Effect of the number of samples on the accuracy of the predicted fields. On the right: variation of the average error on the average concentration in the test set. On the left: variation of the root mean squared error in the parallel and perpendicular direction to flow on the average concentration in the test set. The test set is the same in all the cases.

scalar quantity. Our approach yields a a robust and flexible surrogate model that can be integrated in multiscale modeling workflows.

We studied the effect of different input features to inform the model about the structure of the domain and the boundary conditions, we showed that the Euclidean distance, the time of flight, the pressure drop, and the diffusion coefficient are sufficient to obtain very accurate predictions for a wide range of sphere pack arrangements under varying operative conditions.

This workflow is applicable to a wide range of systems that involve transport and reaction in porous media. The input features we proposed describe uniquely the domain, hence these can be employed in new geometries. The dimension of the domain does not represent a problem for this approach, in fact, it is possible to train MSNet on three-dimensional datasets to create a model able to infer into 3D domains.

We showed how these models can be of use in aiding scale-bridging procedures for multi-scale simulations, or for the prediction of process-scale performance of reaction phenomena determined by complex micro-scale structures.

In conclusion, we demonstrated a methodology that can successfully provide predictions in a split-second of otherwise computationally intensive CFD simulations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code of the multiscale neural network and the dataset are available on GitHub (https://github.com/mulmopro/reactive_ms_net).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cej.2022.140367.

References

[1] A. Singhal, S. Cloete, S. Radl, R. Quinta-Ferreira, S. Amini, Heat transfer to a gas from densely packed beds of monodisperse spherical particles, Chem. Eng. J. 314 (2017) 27–37.
[2] A.H. Thaker, G.M. Karthik, V.V. Buwa, PIV measurements and CFD simulations of the particle-scale flow distribution in a packed bed, Chem. Eng. J. 374 (2019) 189–200.
[3] V. Sassanis, L. Gamet, M. Rolland, R. Ma, V. Pozzobon, Numerical determination of the volumetric heat transfer coefficient in fixed beds of wood chips, Chem. Eng. J. 417 (2021) 128009.
[4] T. Eppinger, K. Seidler, M. Kraume, DEM-CFD simulations of fixed bed reactors with small tube to particle diameter ratios, Chem. Eng. J. 166 (1) (2011) 324–331.
[5] E. Crevacore, G. Boccardo, D. Marchisio, R. Sethi, Microscale colloidal transport simulations for groundwater remediation, Chem. Eng. Trans. 47 (2016) 271–276.
[6] P. Bhuvankar, A. Cihan, J. Birkholzer, Pore-scale CFD simulations of clay mobilization in natural porous media due to fresh water injection, Chem. Eng. Sci. 247 (2022) 117046.
[7] M. Mousavi, M. Prodanovic, D. Jacobi, New classification of carbonate rocks for process-based pore-scale modeling, SPE J. 18 (02) (2013) 243–263.
[8] E. Agostini, G. Boccardo, D. Marchisio, An open-source workflow for open-cell foams modelling: Geometry generation and CFD simulations for momentum and mass transport, Chem. Eng. Sci. 255 (2022) 117583.
[9] C. Yue, Q. Zhang, Z. Zhai, Numerical simulation of the filtration process in fibrous filters using CFD-DEM method, J. Aerosol Sci. 101 (2016) 174–187.
[10] J. Roegiers, S. Denys, CFD-modelling of activated carbon fibers for indoor air purification, Chem. Eng. J. 365 (2019) 80–87.
[11] Y. Xu, X. Zhang, X. Hao, D. Teng, T. Zhao, Y. Zeng, Micro/nanofibrous nonwovens with high filtration performance and radiative heat dissipation property for personal protective face mask, Chem. Eng. J. 423 (2021) 130175.
[12] R. Gautier, T. Dbouk, M.A. Campesi, L. Hamon, J.-L. Harion, P. Pre, Pressure-swing-adsorption of gaseous mixture in isotropic porous medium: Transient 3D modeling and validation, Chem. Eng. J. 348 (2018) 1049–1062.
[13] D.A. Clarke, F. Dolamore, C.J. Fee, P. Galvosas, D.J. Holland, Investigation of flow through triply periodic minimal surface-structured porous media using MRI and CFD, Chem. Eng. Sci. 231 (2021) 116264.
[14] P. Gabrielli, M. Gazzani, M. Mazzotti, The role of carbon capture and utilization, carbon capture and storage, and biomass to enable a net-zero-CO2 emissions chemical industry, Ind. Eng. Chem. Res. 59 (15) (2020) 7033–7045.
[15] H. Chen, M. Yang, C. Huang, Y. Wang, Y. Zhang, M. Zuo, A dynamic model of CO2 diffusion coefficient in shale based on the whole process fitting, Chem. Eng. J. 428 (2022) 131151.
[16] A. Mazzoldi, T. Hill, J.J. Colls, CFD and Gaussian atmospheric dispersion models: A comparison for leak from carbon dioxide transportation and storage facilities, Atmos. Environ. 42 (34) (2008) 8046–8054.
[17] L. Chen, L. Zhang, Q. Kang, H.S. Viswanathan, J. Yao, W. Tao, Nanoscale simulation of shale transport properties using the lattice Boltzmann method: Permeability and diffusivity, Sci. Rep. 5 (1) (2015) 1–8.
[18] G.M. Goldin, A.M. Colclasure, A.H. Wiedemann, R.J. Kee, Three-dimensional particle-resolved models of Li-ion batteries to assist the evaluation of empirical parameters in one-dimensional models, Electrochim. Acta 64 (2012) 118–129.
[19] A.C. Ngandjong, A. Rucci, M. Maiza, G. Shukla, J. Vazquez-Arenas, A.A. Franco, Multiscale simulation platform linking lithium ion battery electrode fabrication process with performance at the cell level, J. Phys. Chem. Lett. 8 (23) (2017) 5966–5972.
[20] L. Zhao, Z. Li, B. Caswell, J. Ouyang, G.E. Karniadakis, Active learning of constitutive relation from mesoscopic dynamics for macroscopic modeling of non-Newtonian flows, J. Comput. Phys. 363 (2018) 116–127.

[21] L. Zhao, Z. Li, Z. Wang, B. Caswell, J. Ouyang, G.E. Karniadakis, Active- and transfer-learning applied to microscale-macroscale coupling to simulate viscoelastic flows, J. Comput. Phys. 427 (2021) 110069.
[22] N. Di Pasquale, J.D. Elliott, P. Hadjidoukas, P. Carbone, Dynamically polarizable force fields for surface simulations via multi-output classification neural networks, J. Chem. Theory Comput. 17 (7) (2021) 4477–4485.
[23] D. Fissore, A.A. Barresi, D. Manca, Modelling of methanol synthesis in a network of forced unsteady-state ring reactors by artificial neural networks for control purposes, Chem. Eng. Sci. 59 (19) (2004) 4033–4041.
[24] S. Whitaker, The Method of Volume Averaging, Vol. 13, Springer Science & Business Media, 2013.
[25] F. Municchi, M. Icardi, Macroscopic models for filtration and heterogeneous reactions in porous media, Adv. Water Resour. 141 (2020) 103605.
[26] G. Allaire, A.-L. Raphael, Homogenization of a convection–diffusion model with reaction in a porous medium, C. R. Math. 344 (8) (2007) 523–528.
[27] U. Hornung, Homogenization and Porous Media, Vol. 6, Springer Science & Business Media, 1996.
[28] I. Battiato, D.M. Tartakovsky, Applicability regimes for macroscopic models of reactive transport in porous media, J. Contam. Hydrol. 120 (2011) 18–26.
[29] M. Elimelech, J. Gregory, X. Jia, Particle Deposition and Aggregation: Measurement, Modelling and Simulation, Butterworth-Heinemann, 2013.
[30] K.E. Nelson, T.R. Ginn, New collector efficiency equation for colloid filtration in both natural and engineered flow conditions, Water Resour. Res. 47 (5) (2011).
[31] W.P. Johnson, M. Hilpert, Upscaling colloid transport and retention under unfavorable conditions: Linking mass transfer to pore and grain topology, Water Resour. Res. 49 (9) (2013) 5328–5341.
[32] G. Boccardo, D.L. Marchisio, R. Sethi, Microscale simulation of particle deposition in porous media, J. Colloid Interface Sci. 417 (2014) 227–237.
[33] I.L. Molnar, W.P. Johnson, J.I. Gerhard, C.S. Willson, D.M. O'carroll, Predicting colloid transport through saturated porous media: A critical review, Water Resour. Res. 51 (9) (2015) 6804–6845.
[34] H. Ma, W.P. Johnson, Colloid retention in porous media of various porosities: Predictions by the hemispheres-in-cell model, Langmuir 26 (3) (2010) 1680–1687.
[35] H. Ma, M. Hradisky, W.P. Johnson, Extending applicability of correlation equations to predict colloidal retention in porous media at low fluid velocity, Environ. Sci. Technol. 47 (5) (2013) 2272–2278.
[36] F. Messina, T. Tosco, R. Sethi, On the failure of upscaling the single-collector efficiency to the transport of colloids in an array of collectors, Water Resour. Res. 52 (7) (2016) 5492–5505.
[37] M. Saeedan, A.R.S. Nazar, Y. Abbasi, R. Karimi, CFD investigation and neutral network modeling of heat transfer and pressure drop of nanofluids in double pipe helically baffled heat exchanger with a 3-D fined tube, Appl. Therm. Eng. 100 (2016) 721–729.
[38] Y. Ding, Y. Zhang, Y.M. Ren, G. Orkoulas, P.D. Christofides, Machine learning-based modeling and operation for ALD of SiO2 thin-films using data from a multiscale CFD simulation, Chem. Eng. Res. Des. 151 (2019) 131–145.
[39] A. Marcato, G. Boccardo, D.L. Marchisio, A computational workflow to study particle transport in porous media: Coupling CFD and deep learning, in: Computer Aided Chemical Engineering, vol. 48, Elsevier, 2020, pp. 1759–1764.
[40] A. Marcato, G. Boccardo, D. Marchisio, A computational workflow to study particle transport and filtration in porous media: Coupling CFD and deep learning, Chem. Eng. J. 417 (2021) 128936.
[41] J. Wu, X. Yin, H. Xiao, Seeing permeability from images: Fast prediction with convolutional neural networks, Sci. Bull. 63 (18) (2018) 1215–1222.
[42] N. Alqahtani, F. Alzubaidi, R.T. Armstrong, P. Swietojanski, P. Mostaghimi, Machine learning for predicting properties of porous media from 2D X-ray images, J. Pet. Sci. Eng. 184 (2020) 106514.
[43] A. Marcato, G. Boccardo, D. Marchisio, From computational fluid dynamics to structure interpretation via neural networks: An application to flow and transport in porous media, Ind. Eng. Chem. Res. (2022).
[44] Y.D. Wang, T. Chung, R.T. Armstrong, P. Mostaghimi, ML-LBM: Predicting and accelerating steady state flow simulation in porous media with convolutional neural networks, Transp. Porous Media 138 (1) (2021) 49–75.
[45] J.E. Santos, D. Xu, H. Jo, C.J. Landry, M. Prodanović, M.J. Pyrcz, PoreFlow-Net: A 3D convolutional neural network to predict fluid flow through porous media, Adv. Water Resour. 138 (2020) 103539.
[46] O. Hennigh, Lat-Net: Compressing lattice Boltzmann flow simulations using deep neural networks, 2017, arXiv preprint arXiv:1705.09036.
[47] N. Wang, H. Chang, D. Zhang, Efficient uncertainty quantification for dynamic subsurface flow with surrogate by theory-guided neural network, Comput. Methods Appl. Mech. Engrg. 373 (2021) 113492.
[48] E. Laloy, R. Hérault, J. Lee, D. Jacques, N. Linde, Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network, Adv. Water Resour. 110 (2017) 387–405.
[49] S. Mo, Y. Zhu, N. Zabaras, X. Shi, J. Wu, Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media, Water Resour. Res. 55 (1) (2019) 703–728.
[50] M. Tang, Y. Liu, L.J. Durlofsky, A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems, J. Comput. Phys. 413 (2020) 109456.
[51] S. Mo, N. Zabaras, X. Shi, J. Wu, Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification, Water Resour. Res. 55 (5) (2019) 3856–3881.
[52] J.E. Santos, Y. Yin, H. Jo, W. Pan, Q. Kang, H.S. Viswanathan, M. Prodanović, M.J. Pyrcz, N. Lubbers, Computationally efficient multiscale neural networks applied to fluid flow in complex 3D porous media, Transp. Porous Media 140 (1) (2021) 241–272.
[53] J. Bear, Dynamics of Fluids in Porous Media, Courier Corporation, 1988.
[54] J. Salles, J.-F. Thovert, R. Delannay, L. Prevors, J.-L. Auriault, P.M. Adler, Taylor dispersion in porous media. Determination of the dispersion tensor, Phys. Fluids A 5 (10) (1993) 2348–2376.
[55] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[56] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, RepVGG: Making VGG-style ConvNets great again, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733–13742.
[57] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[58] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.
[59] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[60] S. Ruder, An overview of gradient descent optimization algorithms, 2016, arXiv preprint arXiv:1609.04747.
[61] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 6 (02) (1998) 107–116.
[62] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11976–11986.
[63] J.T. Barron, Continuously differentiable exponential linear units, 2017, arXiv preprint arXiv:1704.07483.
[64] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUS), 2015, arXiv preprint arXiv:1511.07289.
[65] L. Lu, Y. Shin, Y. Su, G.E. Karniadakis, Dying relu and initialization: Theory and numerical examples, 2019, arXiv preprint arXiv:1903.06733.
[66] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, 2015, CoRR arXiv:1505.04597.
[67] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015, CoRR arXiv:1512.03385.
[68] J. Furtney, scikit-fmm: The fast marching method for Python, 2022, https://github.com/scikit-fmm/scikit-fmm. (Online Accessed 2 February 2022).
[69] J.T. Gostick, Z.A. Khan, T.G. Tranter, M.D.R. Kok, M. Agnaou, M. Sadeghi, R. Jervis, PoreSpy: A python toolkit for quantitative analysis of porous media images, J. Open Source Softw. 4 (37) (2019) 1296.